riacs Research Institute for Advanced Computer Science

Design and Control of Large Collections of Learning Agents

Adrian Agogino

RIACS Technical Report 1.07

April 2001

Design and Control of Large Collections of Learning Agents

Adrian Agogino, University of Texas at Austin

RIACS Technical Report 1.07

April 2001

The intelligent control of multiple autonomous agents is an important yet difficult task. Previous methods used to address this problem have proved to be either too brittle, too hard to use, or not scalable to large systems. The "Collective Intelligence" project at NASA/Ames provides an elegant, machine-learning approach to address these problems. This approach mathematically defines some essential properties that a reward system should have to promote coordinated behavior among reinforcement learners. My work has focused on creating additional key properties and algorithms within the mathematics of the Collective Intelligence framework. One of the additions will allow agents to learn more quickly, in a more coordinated manner. The other will let agents learn with less knowledge of their environment. These additions will allow the framework to be applied more easily, to a much larger domain of multi-agent problems.

This work was supported in part by the National Aeronautics and Space Administration under Cooperative Agreement NCC 2-1006 with the Universities Space Research Association (USRA).

Contents

Chapter 1

Completed Research

The collective intelligence framework (COIN) [WTar, WTF99, WWT00, WWT99] exhibits tremendous flexibility and promise as a framework for controlling multi-agent systems. To take immediate advantage of the benefits of COIN it is necessary to map the theory to some of the fundamental domains of the reinforcement learning community. These domains also serve as a test bed that can to be used to solve many COIN issues related to macrolearning and communication restrictions. One important domain is the learning of sequence of actions to optimize a long-term reward. The multi-agent case adds additional complexity as agents have to learn actions that contribute to a long-term utility of all agents. I have addressed this problem by showing how certain rewards can be used, and have proved that they are factored and exhibit high learnability within the COIN framework. To demonstrate their effectiveness, I have used them in the domains of Mars rovers, and predator/prey interaction.

1.1 Collective Learning of Sums of Rewards

A common reinforcement learning problem is when a learner tries to optimize a sum of discounted rewards over time. In an episodic task with τ time steps this sum is: $\sum_{t=0}^{\tau} \lambda^t r_t$, where r_t is the reward received at time step t and λ is the discounting factor. A common example of this problem is "Grid World" [SB98] where an agent has to find a path through a grid leading to a goal (Figure 1.1). A variant on Grid World is the Mars Rover Problem, where rovers search a grid for interesting Martian rocks.

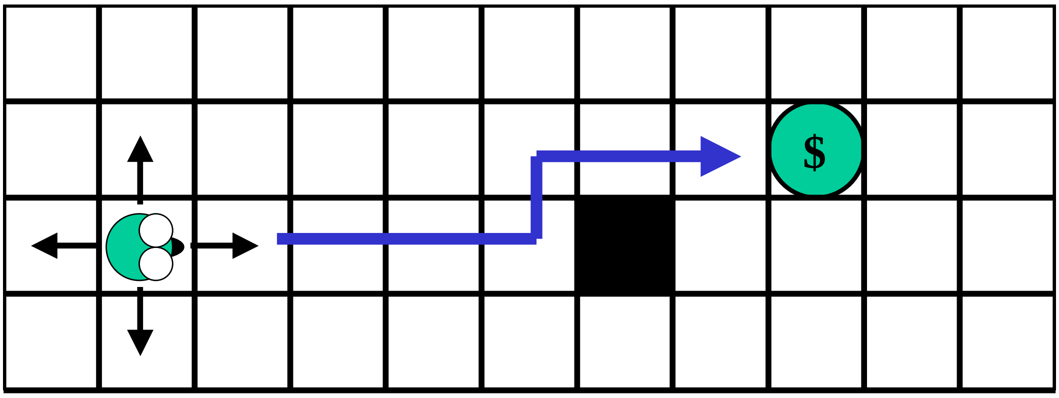

Figure 1.1: Grid World: An agent learns a path towards a goal

1.1.1 Mars Rover Problem

In the Mars Rover Problem (Figure 1.2) a group of m rovers moves around on an n by n grid for τ time steps. While moving, the rovers observe interesting rocks of various values scattered around the grid. The global objective is to discover the highest total value of rocks. The state of the rover at time step t is the grid location of the rover at that time step. The actions of the rovers are to move up, down, left and right. At the end of τ time steps each rover receives τ rewards corresponding to its actions. Each rover has a reinforcement learner that tries to optimize a sum of rewards.

Suppose that the reward were the value of the rock observed as a result of an action. Then a single agent system with a competent reinforcement learner would be able to find a path that would maximize the value of the rewards. In contrast in a multi-agent environment, a high value of the world utility may not be achieved with this same reward. This will happen when the world utility does not benefit from a rock being observed more than once. In this case several rovers will converge on the most valuable rock, while a better solution would be for the rovers to distribute their observations across the grid.

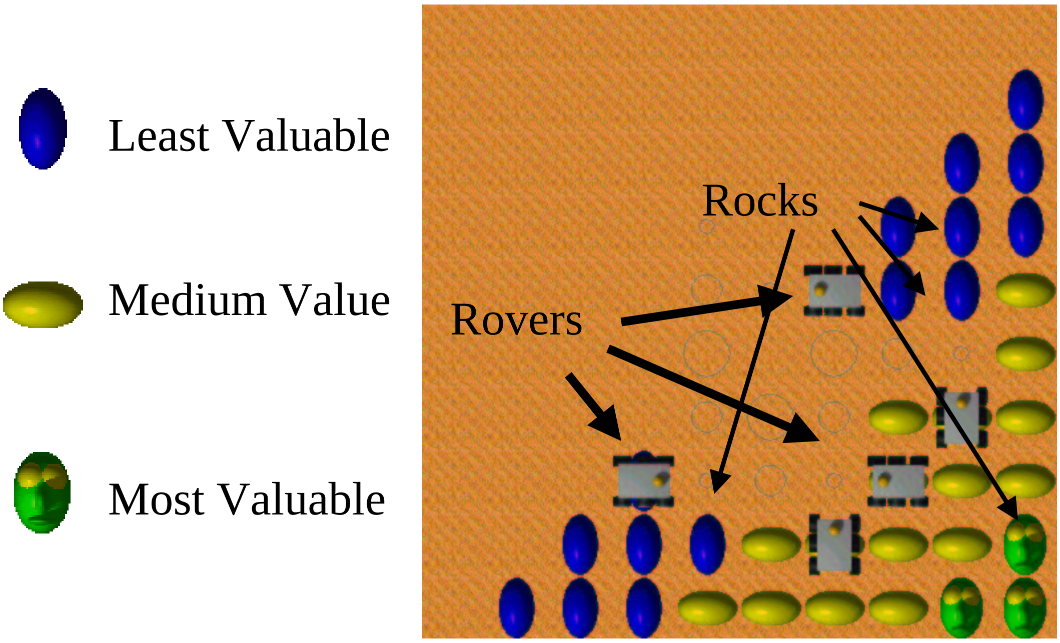

Figure 1.2: Rovers move in four directions across virtual world, observing rocks.

To find rewards to promote the rovers to work together, COIN theory can be applied. To do this the world line, $\underline{\zeta}$, and global utility, $G(\underline{\zeta})$, must be first be defined. In one useful definition, $\underline{\zeta}$ contains the rovers' locations. $G(\underline{\zeta})$ then computes all the rocks that have been observed based on the rovers' locations. Within an episode the effect set for a rover is just its own locations, since rovers do not influence each others locations. The Wonderful Life Utility for an episode can now be defined as: $WLU_\eta(\underline{\zeta}) = G(\underline{\zeta}) - G(CL_{S^{eff}_\eta}(\underline{\zeta}))$. Unfortunately this $WLU_\eta(\underline{\zeta})$ is not useful as a reward value, since it is over the entire episode, while the rovers want a reward value for every action. A single time step WLU can be defined by defining a single time step global utility, $G_t(\underline{\zeta})$ as a computation of all the rocks that were observed at a single time step. The single time step WLU is then: $WLU_{\eta,t}(\underline{\zeta}) = G_t(\underline{\zeta}) - G_t(CL_{S^{eff}_\eta}(\underline{\zeta}))$.

Theorem: When $WLU_{\eta,t}(\underline{\zeta})$ is used as a single time step reward, a learner optimizing an undiscounted sum of rewards ($\lambda = 1$) will optimize the full $WLU_\eta(\underline{\zeta})$.

Proof:

$$\begin{aligned}
\sum_t r_t &= \sum_t WLU_{\eta,t}(\underline{\zeta}) \\
&= \sum_t (G_t(\underline{\zeta}) - G_t(CL_{eff_\eta}(\underline{\zeta}))) \\
&= \sum_t G_t(\underline{\zeta}) - \sum_t G_t(CL_{eff_\eta}(\underline{\zeta})) \\
&= G(\zeta) - G(CL_{eff_\eta}(\underline{\zeta})) \\
&= WLU_\eta(\underline{\zeta})
\end{aligned}$$

To evaluate the effectiveness of COIN methods with the Mars rover problem experiments were run where the rovers were given three different types of rewards. The first reward was the selfish reward where each rover received credit for the value of a rock when it was on the same grid space as the rock. This reward represents the amount of information that would be gained by the rover if no other rovers where there. This would probably be the optimal reward choice in a single agent environment. The second reward is the single time step global utility, G_t. This reward represents all the information gained at a time step. The last reward is the single time step WLU, which represents the contribution a rover made to the total information gain.

Figure 1.3 shows the results when there are 10 agents taking actions for 10 times steps in a 10x10 grid. The selfish agents actually did worse than random because they all went for the same rocks. The agents in the team game learned very slowly and could only do slightly better than random. In contrast the agents using the WLU quickly performed close to optimal. Figure 1.4 shows that the WLU can scale, as it performs well even when there are 100 rovers.

1.1.2 Alternative Reinforcement Learners

Learning to optimize a sum of values over time can also be done with simple reinforcement learners that only optimize their immediate reward, if the reward is chosen properly. Suppose in a single rover environment that there is one low valued rock next to the rover and that there is a high valued rock further away in the opposite direction. If the reward value is simply the value of the rock then the rover will always go for the low valued rock, since it is optimizing its immediate reward. In contrast if the reward value at time t' is the sum of the values of all the rocks received for the remainder of the episode, $\sum_{t \geq t'} G_t(\underline{\zeta})$, then the rover will go for the high valued rock. This solution is known as Monte-Carlo learning. Interestingly a simple application of COIN theory will lead to the same solution in the single agent case.

Theorem: A single agent system using WLU with a simple learner employes Monte-Carlo learning.

Proof: The WLU can be computed using the same definition of global utility as in the previous problem. The effect set for an action contains all the future states of the agent. The WLU for an action at time t' is then:

$$\begin{aligned}
G(\underline{\zeta}) - G(CL_{S_\eta^{eff}}(\underline{\zeta})) &= \sum_t G_t(\underline{\zeta}) - \sum_t G_t(CL_{eff_\eta}(\underline{\zeta})) \\
&= \sum_t G_t(\underline{\zeta}) - (\sum_{t<t'} G_t(CL_{eff_\eta}(\underline{\zeta})) + \sum_{t \geq t'} G_t(CL_{eff_\eta}(\underline{\zeta})))
\end{aligned}$$

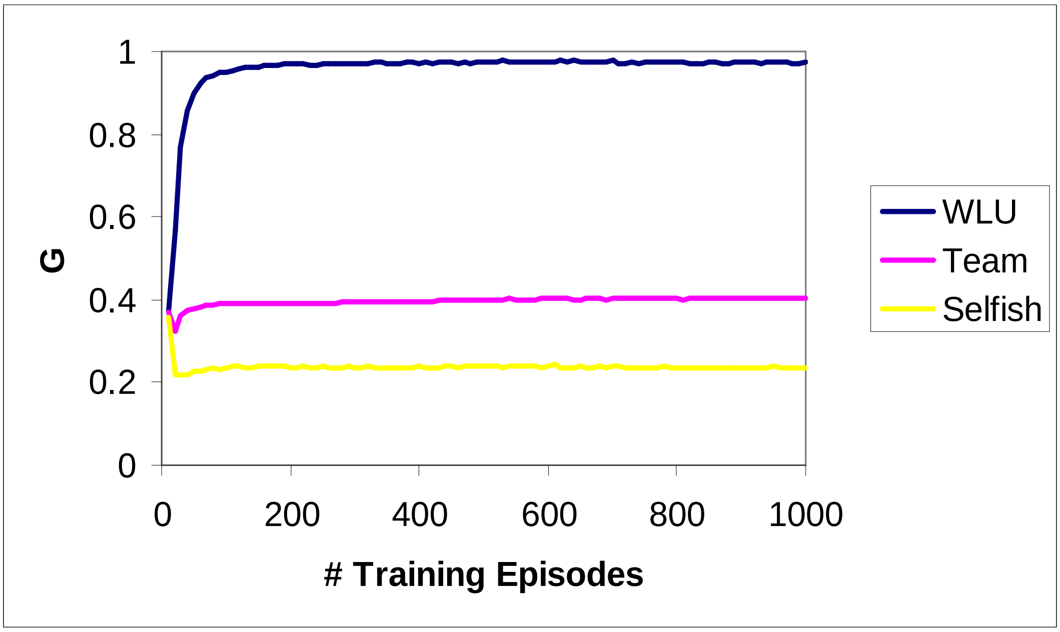

Figure 1.3: Global performance of different utility functions (10 rovers on a 10x10 grid).

$$\begin{aligned} &= \sum_{t} G_t(\underline{\zeta}) - (\sum_{t<t'} G_t(\underline{\zeta}) + \sum_{t\geq t'} G_t(CL_{eff_\eta}(\underline{\zeta})))) \\ &= \sum_{t\geq t'} (G_t(\underline{\zeta}) - G_t(CL_{eff_\eta}(\underline{\zeta}))) \end{aligned}$$

When there is only one agent, $G_t(CL_{eff_\eta}(\underline{\zeta}))$ is zero so the WLU is $\sum_{t\geq t'} G_t(\underline{\zeta})$: Monte-Carlo learning.

While there is no clear definition of Monte-Carlo learning for multi-agents systems the WLU provides a good solution. The WLU will measure the agents contribution to the future global utility of the system. Note that this is different from the WLU for Q-learning which measures an agent's contribution for a single time step.

On certain tests, the performance of the rovers using simple learners was equal to the ones using Q-learners. Also there is a possibility of improving the performance of the simple learners through macrolearning of effect sets. For example if current actions do not significantly affect the states of a rover further than x time steps into the future, the effect set only has to include elements for x time steps. Using the WLU the learner would only have to learn how to optimize for x time steps into the future instead of to the end of the episode. If x is much smaller than the remaining time left in an episode then learning will be much faster. Figure 1.5 shows that performance can be significantly improved when a smaller effect set is used.

It is also easy to add general prior information to simple learners that use a table lookup to determine their actions. Prior knowledge can be added simply by biasing their table upon initialization. For example in the Mars Rover Problem if the designer has a general idea of where the valuable rocks are, the table can be modified to set the rover in the right general direction. Figure 1.6 shows that learning can be significantly speeded up when this is done. In general this can also be done with a Q-learner, but much more care has to be taken since the learning of one table depends on the contents of other tables.

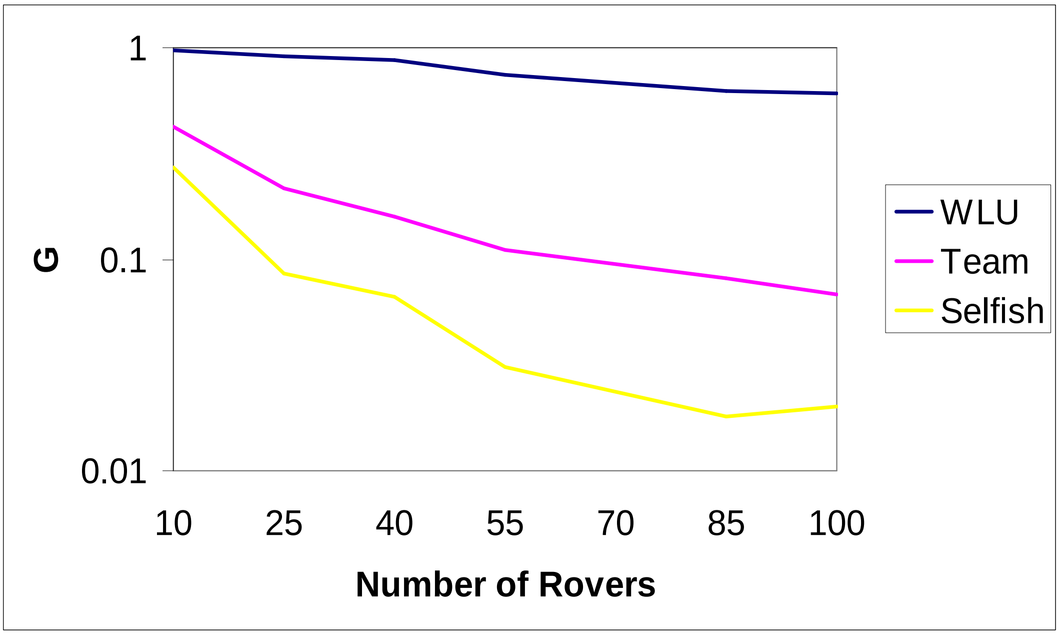

Figure 1.4: Performance with different number of rovers.

1.2 Continuous State Learners

In previous research COIN theory has been applied to table lookup learning methods, which are only useful in domains that have a relatively small number of states. In many real world domains the state space is often continuous so table based learning cannot be used. Instead other machine learners, such as neural nets must be employed. While continuous learning systems typically do not always converge they still have been shown to useful in many domains. This section will show how it is possible to apply COIN to improve performance in continuous domains.

1.2.1 Predator/Prey with Goal Problem

In this problem, a population of "prey" start from a fixed location and try to travel to a single goal. When the prey arrives at the goal it is returned to the starting position. The global utility of the system is the total number of times a prey makes it to a goal. While they are traveling a single "predator" tries to block their path. To navigate the prey can observe the approximate distance and direction of the goal and predator. The distance measure is continuous making the state space continuous as well. While the space can be discretized, a reasonable approximation would yield 10^8 states.

To navigate, each prey employs a single neural network. The input to the neural network is its observations, and the output is the direction and the distance it should move. Learning is implemented through a simulated annealing method. The general principle of this method is that the neural networks are mutated until the prey achieve high performance. Specifically if a mutation results in a higher personal utility then the mutation is kept. Otherwise a decision is made as to whether to keep the mutation based on a Boltzmann distribution. The rate of mutation is lowered throughout training.

The personal utilities is generated by measuring the performance of the system for τ time steps after a mutation. The definition of the WLU depends on the definition of $\underline{\zeta}$. In the first definition $\underline{\zeta}$ contains

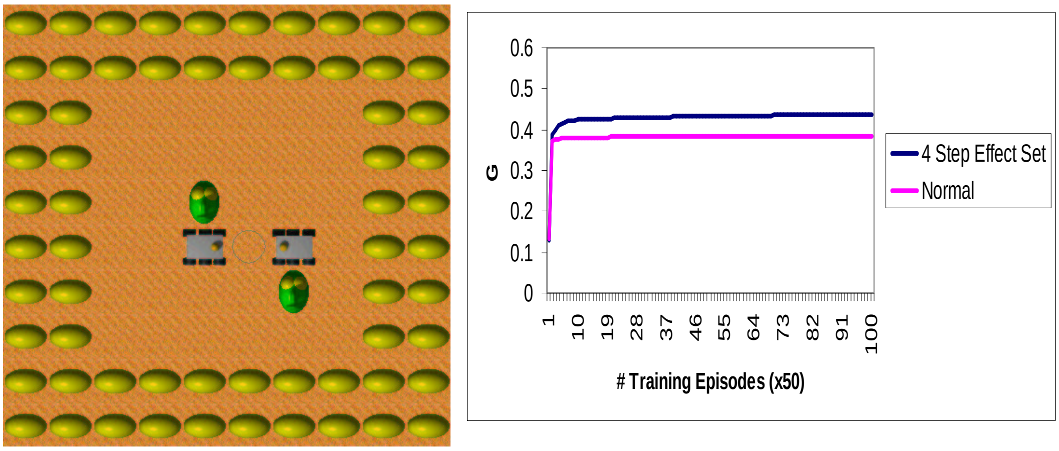

Figure 1.5: Reduction of effect set can increase performance.

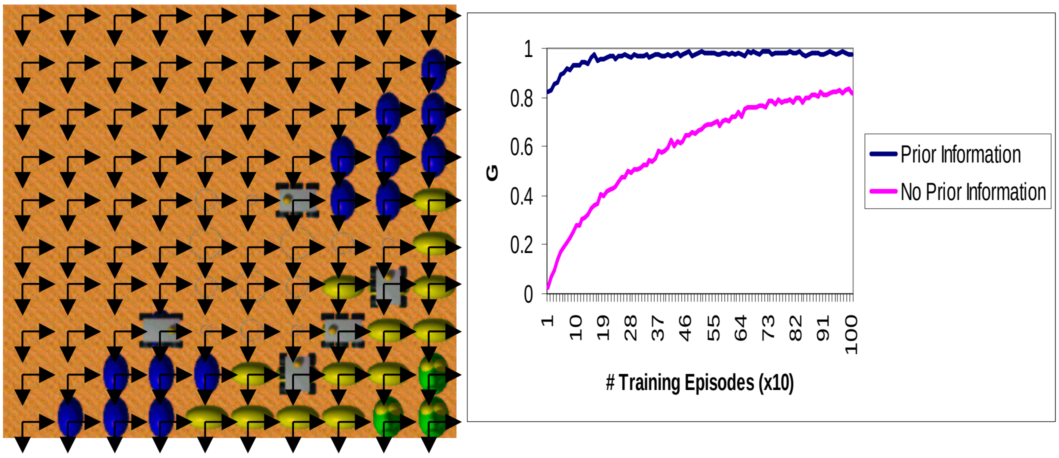

Figure 1.6: Adding prior knowledge significantly speeds up learning.

the locations of the prey. The global utility is then computed on whether the prey made it to the goal. In this case the effect set for one of the prey contains the locations of all the other prey. This large effect set is needed since even though the prey cannot observe each other, they can observe the predator which can observe all the prey. Since the effect set includes the entire space, this first definition of $\underline{\zeta}$ produces a WLU that is equivalent to the team game.

In another definition, $\underline{\zeta}$ contains the weights in the neural networks of the prey. The global utility is then computed by simulating the system based on the weights and determine the total number of times the prey reached a goal. In this case the effect set for one of the prey only includes itself, since the weights do not effect each other during a testing episode. The WLU will therefore be the agents own contribution to the system and will be much more learnable than with the previous definition of $\underline{\zeta}$.

The two versions of WLU were tested against a greedy utility. This greedy utility was the number of times one of the prey arrived at the goal. The system employing the greedy utility performed well early in training, but did worse as they learned the most direct path to the goal (Figure 1.7, left). When all the prey converged on the same path, the predator could block them all. This illustrated the classic problem of increasingly competent learners causing the overall performance to go down. While the system using the WLU with the first definition of $\underline{\zeta}$ performed better, the prey could not learn effectively in the team game. The system using the second definition of $\underline{\zeta}$ performed the best as some of the prey learned to sacrifice themselves to distract the predator (Figure 1.7, right) .

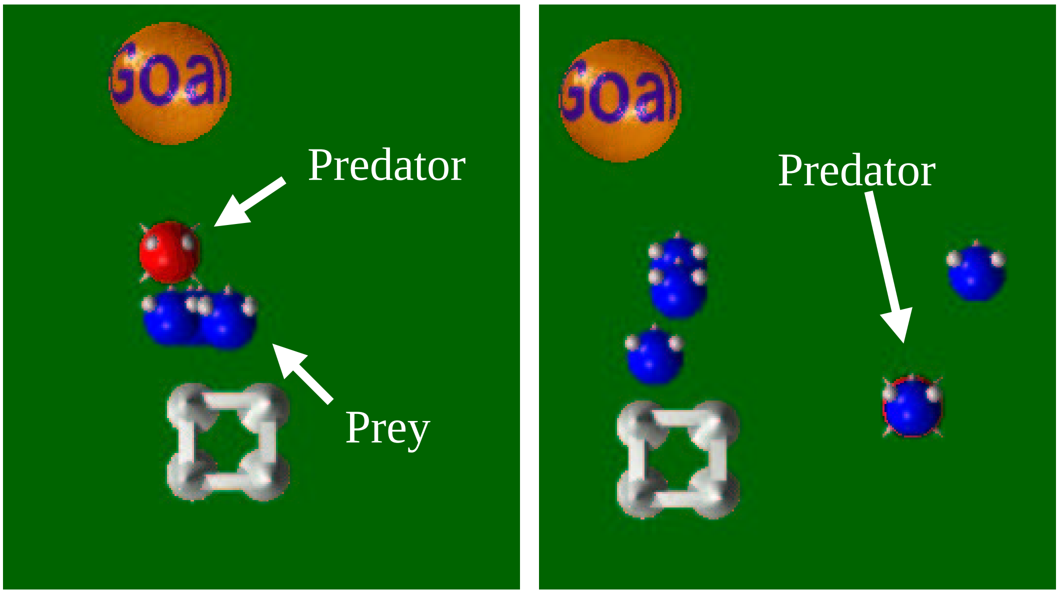

Figure 1.7: Selfish learners (left figure) suffer when enemy is able to block them all. Using WLU (right figure) some prey learn to distract enemy to help others.

Bibliography

[SB98] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.

[WTF99] D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press, 1999.

[WTar] D. H. Wolpert and K. Tumer. An Introduction to Collective Intelligence. In J. M. Bradshaw, editor, *Handbook of Agent technology*. AAAI Press/MIT Press, to appear. Available as tech. rep. NASA-ARC-IC-99-63 from `http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`.

[WWT99] D. H. Wolpert, K. Wheeler, and K. Tumer. General principles of learning-based multi-agent systems. In *Proceedings of the Third International Conference of Autonomous Agents*, pages 77–83, 1999.

[WWT00] D. H. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6), March 2000.